1 General/Finance/Statistics

Program Library

Percentage
Metric System
Memory
Games
Dates
Finance
Mortgages
Statistics

Sinclair Cambridge Programmable

Printed by Hobsons Press (Cambridge) Ltd

General / Finance / Statistics

1

CONTENTS

STATISTICS

How to use these programs

Each program is arranged as follows:

1. On the left of the page, explanatory information and the 'execution sequence', the sequence of keystrokes necessary for running the program. Results displayed are printed in gold.
2. In the first column on the right hand side of the page, the sequence of keystrokes which make up the program.
3. In the second and third columns on the right hand side of the page, the program in check symbol and step number form (see section on checking the program).

Notes

1. Where a key has more than one function, the relevant function is printed as the keystroke in the first column

   e.g. the keystroke [cos / 8 / arccos] may appear as 8, cos or arccos.
2. The symbol ▾ within a program always refers to the key [▾ ·/EE/–]
3. The symbol # refers to [ChN/# 3]
4. The abbreviation gin is 'go if neg' and so refers to the key [1 go if neg]

Entering the program

To enter a program into the calculator:

1.	Press [▲▼] [▲▼] [2 go to] [0] [0]	Display shows step programmed at 00 in check symbol form as described below.
2.	Press [▲▼] [learn RUN]	No change in display.
3.	Press the sequence of keys for the program as shown in the first column of the program page.	At each stage the step about to be overwritten is displayed. When the machine is first switched on every step is zero.
4.	Press [C/CE]	Normal number display is resumed.
5.	Press [▲▼] [▲▼] [2 go to] [0] [0]	The step programmed at 00 will be displayed.

Checking the program

Each of the programs in the library is shown in check symbol form in the second column on the right-hand side of the page.

Press [▲▼] [C/CE] (step) repeatedly, and at each stage the check symbol will appear on the left of the display with the step number on the right. Ignore the four zeros in the display.

e.g. A.0000 03 (check symbol: A; step number: 03)

After stepping through the program, press

[▲▼] [▲▼] [2] (go to) [0] [0] before execution.

Finally, press [C/CE] and the program is ready for use.

Correcting the program

If the check symbol for a particular step number is not as indicated in the last two columns of the program page:

1. Press [▲▼] [▲▼] [2] (go to) followed by the step number if the appropriate step number is not already displayed.
2. Press [▲▼] [RUN] (learn)
3. Enter the correct keystroke. The display will then show the next step in the program. If this is also incorrect, enter the correct keystroke. At each stage, the step about to be overwritten will be displayed.
4. When correction has been completed, press [C/CE]. Any step which has not been overwritten will not be affected.
5. Press [▲▼] [▲▼] [2] (go to) [0] [0]

Note
To restore normal use of the calculator after entering or checking the program, press [C/CE]

Running the program

Press the sequence of keys as shown in the program library in the execution sequence. Results displayed are printed in gold.

POWERS

ln	4	00
X	·	01
stop	0	02
=	–	03
▼	A	04
e^x	4	05
stop	0	06
▼	A	07
goto	2	08
0	0	09
0	0	10
		11
		12
		13
		14
		15
		16
		17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

To find x^y

Execution:

x / RUN / y / RUN / x^y $\qquad x > 0$

This program can be used inside parentheses and does not affect memory.

ROOTS

To find the yth root of x

Execution:

x / RUN / y / RUN / $\sqrt[y]{x}$

ln	4	00
÷	G	01
stop	0	02
=	–	03
▼	A	04
e^x	4	05
stop	0	06
▼	A	07
goto	2	08
0	0	09
0	0	10
		11
		12
		13
		14
		15
		16
		17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

PERCENTAGE FUNCTIONS

(	6	00
X	·	01
stop	0	02
÷	G	03
#	3	04
1	1	05
0	0	06
0	0	07
=	–	08
)	6	09
stop	0	10
▼	A	11
goto	2	12
0	0	13
0	0	14
		15
		16
		17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

Execution:

1. x / = / RUN / a / RUN / a% of x
2. / x / + / RUN / a / RUN / a% of x / = / x + a% of x
3. / x / – / RUN / a / RUN / a% of x / = / x – a% of x /

MEMORY FUNCTIONS

Memory contains y initially:

Execution:

M+: x / RUN / + / RUN / (x in display, x + y in memory)

M−: x / RUN / − / RUN / (x in display, y − x in memory)

MX: x / RUN / X / RUN / (x in display, x y in memory)

M÷: x / RUN / ÷ / RUN / (x in display, y ÷ x in memory)

MC: x / RUN / C/CE / C/CE / X / RUN / (x in display, 0 in memory)

STO−: x / RUN / C/CE / C/CE / − / RUN / (x in display, −x in memory)

In each case, the original contents y of the memory are displayed after the first / RUN /.

▼	A	00
MEx	5	01
stop	0	02
rcl	5	03
=	−	04
▼	A	05
MEx	5	06
stop	0	07
▼	A	08
goto	2	09
0	0	10
0	0	11
		12
		13
		14
		15
		16
		17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

HOLDING AN EXTRA CONSTANT IN PROGRAM MEMORY

Suppose there is an extra number you want to store while doing calculations, for example the velocity of light

$c = 2{\cdot}997925 \times 10^8 \ \ \mathrm{m\,s^{-1}}$.

The number can be stored in the program memory as shown opposite.

Each time you need to use the constant, just press / RUN /. This will enter the constant and complete the last operation, just like the sequence / ▲▼ / rcl / = / if the constant were stored in the memory. However, the memory can still be used to store other numbers, and the program will also operate inside parentheses.

This idea can be extended to store several constants if required.

#	3	00
2	2	01
·	A	02
9	9	03
9	9	04
7	7	05
9	9	06
2	2	07
5	5	08
·	A	09
8	8	10
=	–	11
stop	0	12
▼	A	13
goto	2	14
0	0	15
0	0	16
		17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

HOLDING TWO EXTRA CONSTANTS IN PROGRAM MEMORY

#	3	00
1	1	01
·	A	02
0	0	03
7	7	04
4	4	05
8	8	06
3	3	07
2	2	08
1	1	09
=	–	10
stop	0	11
▼	A	12
goto	2	13
0	0	14
0	0	15
#	3	16
4	4	17
·	A	18
3	3	19
8	8	20
6	6	21
5	5	22
7	7	23
9	9	24
=	–	25
stop	0	26
▼	A	27
goto	2	28
0	0	29
0	0	30
		31
		32
		33
		34
		35

The exact way this is done depends on the way that the constants will be needed.

1. One constant readily accessible, the other a little more difficult to recover

To use the const. 1·0748321 just press / RUN /

To use the const. 4·386579 press
▲▼ / ▲▼ / goto / 1 / 6 / RUN

This program can be used inside parentheses and does not affect normal memory use.

HOLDING TWO EXTRA CONSTANTS IN PROGRAM MEMORY

2. Constants wanted alternately

Pressing / RUN / will recall constants alternately.

To recover a constant out of turn press
▲▼ / ▲▼ / goto / 0 / 0 / RUN / for 1·0748321

and
▲▼ / ▲▼ / goto / 1 / 2 / RUN / for 4·386579

(If the second constant is wanted at the beginning of a calculation then / RUN / RUN / will work too.)

This program can be used inside parentheses and does not affect normal memory use.

#	3	00
1	1	01
·	A	02
0	0	03
7	7	04
4	4	05
8	8	06
3	3	07
2	2	08
1	1	09
=	–	10
stop	0	11
#	3	12
4	4	13
·	A	14
3	3	15
8	8	16
6	6	17
5	5	18
7	7	19
9	9	20
=	–	21
stop	0	22
▼	A	23
goto	2	24
0	0	25
0	0	26
		27
		28
		29
		30
		31
		32
		33
		34
		35

HOLDING TWO EXTRA CONSTANTS IN PROGRAM MEMORY

3. Either constant to be used repeatedly

Operation:

/ RUN / recalls first constant whenever needed until first recall of second constant.

For first recall of second constant:
▲▼ / ▲▼ / goto / 1 / 6 / RUN /
Subsequent / RUN / will recall second constant.

To recall first constant again press
▲▼ / ▲▼ / goto / 0 / 0 / RUN /

#	3	00
1	1	01
·	A	02
0	0	03
7	7	04
4	4	05
8	8	06
3	3	07
2	2	08
1	1	09
=	–	10
stop	0	11
▼	A	12
goto	2	13
0	0	14
0	0	15
#	3	16
4	4	17
·	A	18
3	3	19
8	8	20
6	6	21
5	5	22
7	7	23
9	9	24
=	–	25
stop	0	26
▼	A	27
goto	2	28
1	1	29
6	6	30
		31
		32
		33
		34
		35

STORING THREE OR MORE CONSTANTS IN PROGRAM MEMORY

As an example, three important physical constants which are often associated are stored in the program opposite, namely:

T_0 = absolute temperature of 0°C = 273·152K

k = Boltzmann's constant
= $1{\cdot}380622 \times 10^{-23}$ J K^{-1}

q = electronic charge = $1{\cdot}6021917 \times 10^{-19}$C

For example, to calculate the current in a diode from

$$I = I_s\left(\exp\left(\frac{qV}{kT}\right) - 1\right)$$

where V is the applied voltage, T the junction temperature and I_s the saturation current, use pre-execution:

/ ▲▼ / ▲▼ / goto / 0 / 0 /

Execution:

T / + / RUN / X / RUN / ÷ / RUN / ÷ / X / V / = / ▲▼ / ▲▼ / e^x / – / 1 / X / I_s / = / I

with T in °C and V in volts.

For repeated execution, I_s could be stored in memory.

#	3	00
2	2	01
7	7	02
3	3	03
·	A	04
1	1	05
5	5	06
=	–	07
stop	0	08
#	3	09
1	1	10
·	A	11
3	3	12
8	8	13
0	0	14
6	6	15
2	2	16
·	A	17
·	A	18
2	2	19
3	3	20
=	–	21
stop	0	22
#	3	23
1	1	24
·	A	25
6	6	26
0	0	27
2	2	28
2	2	29
·	A	30
·	A	31
1	1	32
9	9	33
=	–	34
stop	0	35

The constants can be recalled out of order by using the pre-execution:

/ ▲▼ / ▲▼ / goto / 0 / 9 /   for k

/ ▲▼ / ▲▼ / goto / 2 / 3 /   for q

or

/ ▲▼ / ▲▼ / goto / 0 / 0 /   for T_0

This idea can be adapted to store three 9-digit numbers, four 6-digit numbers, five 4-digit numbers, etc., the decimal point counting as a digit. Use / = / steps to fill the remaining spaces, or / ▼ / goto / 0 / 0 / etc. if there is room.

LOGARITHMS TO BASE A

If base is not to be kept the same

ln	4	00
sto	2	01
stop	0	02
ln	4	03
÷	G	04
rcl	5	05
=	–	06
stop	0	07
▼	A	08
goto	2	09
0	0	10
0	0	11
		12
		13
		14
		15
		16
		17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

Execution:

a / RUN / x / RUN / $\log_a x$

CONVERSIONS

Degrees Fahrenheit to degrees Centigrade

Execution:

°F / RUN / °C

–	F	00
#	3	01
3	3	02
2	2	03
÷	G	04
#	3	05
1	1	06
·	A	07
8	8	08
=	–	09
stop	0	10
▼	A	11
goto	2	12
0	0	13
0	0	14
		15
		16
		17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

LOGARITHMS TO BASE A

If the same base is to be used repeatedly

If the same base is to be used repeatedly

Execution:

a / RUN / x_1 / RUN / $\log_a x_1$ / x_2 / RUN / $\log_a x_2$ / · · ·

To set a new base:

▲▼ / ▲▼ / goto / 0 / 0 / a' / RUN / · · · etc.

ln	4	00
sto	2	01
stop	0	02
ln	4	03
÷	G	04
rcl	5	05
=	–	06
▼	A	07
goto	2	08
0	0	09
2	2	10
		11
		12
		13
		14
		15
		16
		17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

CONVERSIONS

Degrees Centigrade to degrees Fahrenheit

X	·	00
#	3	01
1	1	02
·	A	03
8	8	04
+	E	05
#	3	06
3	3	07
2	2	08
=	–	09
stop	0	10
▼	A	11
goto	2	12
0	0	13
0	0	14
		15
		16
		17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

Execution:

°C / RUN /

CONVERSIONS

Feet and inches to metres

X	·	00
#	3	01
1	1	02
2	2	03
+	E	04
stop	0	05
X	·	06
#	3	07
·	A	08
0	0	09
2	2	10
5	5	11
4	4	12
=	—	13
stop	0	14
▼	A	15
goto	2	16
0	0	17
0	0	18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

Execution:

feet / RUN / inches / RUN / metres

Note: 0 must be entered if 0 inches.

CONVERSIONS

Metres to feet and inches

Execution:

metres / RUN / feet / RUN / inches

Note: This program may take some time to execute.

÷	G	00
#	3	01
·	A	02
3	3	03
0	0	04
4	4	05
8	8	06
–	F	07
(	6	08
–	F	09
#	3	10
1	1	11
=	–	12
▼	A	13
gin	1	14
2	2	15
1	1	16
▼	A	17
goto	2	18
0	0	19
9	9	20
+	E	21
#	3	22
1	1	23
=	–	24
sto	2	25
)	6	26
=	–	27
stop	0	28
rcl	5	29
X	·	30
#	3	31
1	1	32
2	2	33
=	–	34
stop	0	35

CONVERSIONS

Pounds and ounces to kilograms

+	E	00
+	E	01
+	E	02
+	E	03
+	E	04
stop	0	05
÷	G	06
#	3	07
3	3	08
5	5	09
·	A	10
2	2	11
7	7	12
4	4	13
=	–	14
stop	0	15
▼	A	16
goto	2	17
0	0	18
0	0	19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

Execution:

lb / RUN / oz / RUN / kg

Note: Enter 0 if 0 oz

CONVERSIONS

Kilograms to pounds and ounces

÷	G	00
#	3	01
·	A	02
4	4	03
5	5	04
3	3	05
6	6	06
–	F	07
(	6	08
–	F	09
#	3	10
1	1	11
=	–	12
▼	A	13
gin	1	14
2	2	15
1	1	16
▼	A	17
goto	2	18
0	0	19
9	9	20
+	E	21
#	3	22
1	1	23
=	–	24
sto	2	25
)	6	26
=	–	27
stop	0	28
rcl	5	29
+	E	30
+	E	31
+	E	32
+	E	33
=	–	34
stop	0	35

Execution:

kg / RUN / [illegible] / RUN / oz

CONVERSIONS

Degrees, minutes, seconds to decimal degrees
Hours, minutes, seconds to decimal hours

+	E	00
(	6	01
stop	0	02
X	·	03
#	3	04
6	6	05
0	0	06
+	E	07
stop	0	08
÷	G	09
#	3	10
3	3	11
6	6	12
0	0	13
0	0	14
=	–	15
)	6	16
=	–	17
stop	0	18
▼	A	19
goto	2	20
0	0	21
0	0	22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

Execution:

deg / RUN / min / RUN / sec / RUN / decimal degrees

or

hr / RUN / min / RUN / sec / RUN / [illegible]

Note: Min and sec must be entered as 0 if zero.

CONVERSIONS

Decimal degrees to degrees, minutes and seconds
Decimal hours to hours, minutes and seconds
Decimal minutes to minutes and seconds

–	F	00
(	6	01
–	F	02
#	3	03
1	1	04
=	–	05
▼	A	06
gin	1	07
1	1	08
4	4	09
▼	A	10
goto	2	11
0	0	12
2	2	13
+	E	14
#	3	15
1	1	16
=	–	17
sto	2	18
)	6	19
=	–	20
stop	0	21
rcl	5	22
X	·	23
#	3	24
6	6	25
0	0	26
=	–	27
stop	0	28
▼	A	29
goto	2	30
0	0	31
0	0	32
		33
		34
		35

Execution:

(i) degrees as decimal / RUN / [illegible] / RUN / RUN / [illegible] / RUN / [illegible]

(ii) hours as decimal / RUN / [illegible] / RUN / RUN / [illegible] / RUN / secs

(iii) minutes as decimal / RUN / [illegible] / RUN / secs

The number of seconds will be shown as a decimal. To use the program again, just enter the new number of degrees, hours or minutes.

In (i) and (ii), after the second RUN the display shows the number of minutes as decimal.

MATCHSTICK GAME

You put N matchsticks down on the table. At each turn, each player may pick up 1, 2, or 3 matchsticks; because you choose the starting number N, the machine has the first turn. The object of the game is to avoid picking up the last matchstick; thus if either player leaves 1 matchstick after his turn he has won.

Execution:

N / RUN / machine plays

/ 1, 2 or 3 / RUN / you play

/ RUN / machine plays

/ 1, 2 or 3 / RUN / you play

etc.

Display each time shows number of matchsticks remaining.

sto	2	00
–	F	01
(	6	02
rcl	5	03
+	E	04
#	3	05
3	3	06
–	F	07
#	3	08
4	4	09
–	F	10
–	F	11
▼	A	12
gin	1	13
0	0	14
7	7	15
+	E	16
▼	A	17
gin	1	18
2	2	19
4	4	20
#	3	21
5	5	22
–	F	23
#	3	24
4	4	25
=	–	26
)	6	27
–	F	28
stop	0	29
=	–	30
stop	0	31
=	–	32
=	–	33
=	–	34
=	–	35

PSEUDO–RANDOM DICE THROWER

This dice is slightly biased, but not too heavily to be convincing!

Execution:

Choose any starting value x between 0 and 1.

x / RUN / d_1 / RUN / d_2 / RUN / d_3 / etc.

where d_1, d_2, d_3 are successive 'throws'.

X	·	00
#	3	01
1	1	02
0	0	03
1	1	04
÷	G	05
#	3	06
1	1	07
7	7	08
+	E	09
(	6	10
—	F	11
+	E	12
#	3	13
1	1	14
=	—	15
▼	A	16
gin	1	17
1	1	18
2	2	19
sto	2	20
)	6	21
=	—	22
stop	0	23
rcl	5	24
▼	A	25
goto	2	26
0	0	27
0	0	28
		29
		30
		31
		32
		33
		34
		35

MOON LANDING GAME

The object of the moon landing game is to land the Lunar Module (LEM) safely on the moon's surface.

The LEM's rocket motor has 'bang-bang' control; in other words it can either be on ('burn') or off ('coast'). Thus the landing consists of a series of burns and coasts of various lengths. Your job is to choose the lengths of these stages. You are of course limited by the amount of fuel on board.

For convenience in programming, the landing is modelled by two programs.

The first program models the first long burn which gets the LEM out of lunar orbit and slows it to a near-vertical descent above the landing site.

The second program models the subsequent series of coasts and burns which should slow the LEM to a soft landing on the moon.

The LEM can withstand landing speeds of up to 5 metres per second. Speeds above this may cause spectacularly disastrous results!

The equations used in the programs are of course only approximate, but the approximations can all be justified.

MOON LANDING GAME

Getting out of orbit

This program computes the final speed, amount of fuel remaining and height after the long initial 'burn'. The initial mass of the LEM, M_0 is 3000kg, including fuel mass F_0 = 2000kg. Orbital speed is 1·7km s^{-1} in close lunar orbit at a height H_0 chosen by the pilot – we suggest 25 to 50km. The rocket motor burns 2kg of fuel per second with an exhaust velocity of 2400m s^{-1}, giving a thrust of 4800N.

The final speed V_1, height H_1, mass M_1 and fuel left F_1 are modelled by:

$$V_1 = V_0 + 2400 \ln\left(\frac{M_0 - 2T}{M_0}\right) \quad \text{m s}^{-1}$$

$$H_1 = \frac{H_0}{2} \quad \text{m}$$

$$F_1 = F_0 - 2T \quad \text{kg}$$

$$M_1 = M_0 - 2T \quad \text{kg}$$

'Burn' time left is given by

$$T_1 = T_0 - T \quad \text{s} \qquad \text{where} \quad T_0 = \frac{F_0}{2} \quad \text{s}$$

Execution:

Choose T and H_0

/ RUN / T / RUN / / RUN /

H_0 / ÷ / 2 / = /

Try different values of T if you wish.

The results from this program are used as starting values for the vertical descent phase.

#	3	00
1	1	01
0	0	02
0	0	03
0	0	04
–	–	05
sto	2	06
stop	0	07
+	E	08
+	E	09
stop	0	10
rcl	5	11
÷	G	12
rcl	5	13
÷	G	14
#	3	15
3	3	16
=	–	17
ln	4	18
X	·	19
#	3	20
2	2	21
4	4	22
0	0	23
0	0	24
+	E	25
#	3	26
1	1	27
7	7	28
0	0	29
0	0	30
=	–	31
stop	0	32
=	–	33
=	–	34
=	–	35

MOON LANDING GAME –

Vertical descent

The exact equations of motion during the vertical descent are modelled by linear approximations using the equations below:

'Burn'

$$F_{i+1} = F_i - 2T_b$$

$$V_{i+1} = V_i + 1{\cdot}6T_b - \frac{4800}{M_{av}} T_b$$

$$H_{i+1} = H_i - V_{av} T_b$$

$$T_{i+1} = T_i - T_b$$

'Coast'

$$F_{i+1} = F_i$$

$$V_{i+1} = V_i + 1{\cdot}6T_c$$

$$H_{i+1} = H_i - V_{av} T_c$$

$$T_{i+1} = T_i$$

where $M_{av} = M_i - T_b = \dfrac{M_i + M_{i+1}}{2}$

and $V_{av} = \dfrac{V_i + V_{i+1}}{2}$

The 'coast' equations are exact, but the 'burn' approximations are less accurate for 'burn' times longer than about 45 seconds. Either choose a succession of shorter 'burn' times or correct V_{i+1} and H_{i+1} as below:

$$V'_{i+1} = V_{i+1} - \frac{400T_b}{F_i + 1000}$$

$$H'_{i+1} = H_{i+1} - \frac{400T_b^2}{F_i + 1000}$$

sto	2	00
+	E	01
stop	0	02
+	E	03
stop	0	04
#	3	05
1	1	06
0	0	07
0	0	08
0	0	09
+	E	10
rcl	5	11
÷	G	12
−	F	13
×	·	14
#	3	15
2	2	16
4	4	17
0	0	18
0	0	19
−	F	20
#	3	21
·	A	22
8	8	23
×	·	24
rcl	5	25
−	F	26
(	6	27
+	E	28
+	E	29
stop	0	30
)	6	31
stop	0	32
×	·	33
rcl	5	34
stop	0	35

Execution:

Decide whether to 'burn' or 'coast' and for how long (T_b or T_c seconds)

Burn: T_i / – / T_b / RUN / T_{i+1} / RUN / F_{i+1} / RUN / V_i / RUN / V_{i+1} / RUN / + / H_i / = / H_{i+1}

Coast: T_c / ▲▼ / sto / ▲▼ / ▲▼ / goto / 2 / 1 / RUN / V_i / RUN / V_{i+1} / RUN / + / H_i / = / H_{i+1}

Tabulate the results as below:

Burn	Coast	Time T_i	Fuel F_i	Speed V_1	Height H_i
*250		750	1500	1262·4416	15000
	3	750	1500	1267·2416	13735·159
10		740	1480	1231·9645	1239·129
	1	740	1480	1233·5645	6·3645

You are now 6 metres above the moon travelling at 1233·5645 metres per sec. Crash!!! Better luck next time!

* using 'getting out of orbit' program.

SUNDAY LETTER 1900 – 2099

Execution:

year / RUN / result

Result	Sunday letter
1	A
2	B
3	C
4	D
5	E
6	F
7	G

To find Easter 1900–2099

Use this program to find the Sunday letter and also find the Golden Number.

Locate the Golden Number in the first column of the Table and read across to find the date of the Paschal Full Moon in the second column.

Read down the third column from the day following the Paschal Full Moon to find the Sunday letter. The date opposite this letter in column 2 is the date of Easter Sunday.

e.g. 1976 Golden number = 1
Sunday letter = C

Column 1 gives Paschal Full Moon as April 14. First C below April 14 is April 18.

Therefore April 18 = Easter Sunday.

–	F	00
#	3	01
2	2	02
1	1	03
0	0	04
7	7	05
÷	G	06
#	3	07
·	A	08
8	8	09
+	E	10
#	3	11
7	7	12
+	E	13
▼	A	14
gin	1	15
1	1	16
1	1	17
(	6	18
–	F	19
+	E	20
#	3	21
1	1	22
=	–	23
▼	A	24
gin	1	25
2	2	26
0	0	27
)	6	28
–	F	29
+	E	30
#	3	31
8	8	32
=	–	33
stop	0	34
=	–	35

GOLDEN NUMBER 1900 – 2099

Execution:

year / RUN / Golden number

Table to find Easter 1900–2099

Golden number	Day and month	Sunday letter
–	March 21	C
14	22	D
3	23	E
–	24	F
11	25	G
–	26	A
19	27	B
8	28	C
–	29	D
16	30	E
5	31	F
–	April 1	G
13	2	A
2	3	B
–	4	C
10	5	D
–	6	E
18	7	F
7	8	G
–	9	A
15	10	B
4	11	C
–	12	D
12	13	E
1	14	F
–	15	G
9	16	A
17	17	B
6	18	C
–	19	D
–	20	E
–	21	F
–	22	G
–	23	A
–	24	B
–	25	C

–	F	00
#	3	01
1	1	02
9	9	03
0	0	04
0	0	05
–	F	06
#	3	07
1	1	08
9	9	09
=	–	10
▼	A	11
gin	1	12
1	1	13
9	9	14
▼	A	15
goto	2	16
0	0	17
6	6	18
+	E	19
#	3	20
2	2	21
0	0	22
=	–	23
stop	0	24
▼	A	25
goto	2	26
0	0	27
0	0	28
		29
		30
		31
		32
		33
		34
		35

DAY OF THE WEEK OF CHRISTMAS DAY (1900 – 2099)

Execution:

year (in full) / RUN / day as a number

where 1 = Sunday
2 = Monday, etc

X	·	00
#	3	01
1	1	02
·	A	03
2	2	04
4	4	05
9	9	06
6	6	07
–	F	08
#	3	09
2	2	10
6	6	11
3	3	12
1	1	13
+	E	14
#	3	15
7	7	16
+	E	17
▼	A	18
gin	1	19
1	1	20
5	5	21
(	6	22
–	F	23
+	E	24
#	3	25
1	1	26
=	–	27
▼	A	28
gin	1	29
2	2	30
4	4	31
)	6	32
=	–	33
stop	0	34
=	–	35

		00
		01
		02
		03
		04
		05
		06
		07
		08
		09
		10
		11
		12
		13
		14
		15
		16
		17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

DISCOUNT

Discounts a series of prices by a given percentage.

Execution:

percentage discount / RUN / gross price / RUN / discounted price / gross price / RUN / discounted price /

To enter a new discount:

▲▼ / ▲▼ / goto / 0 / 0 / new discount / RUN /

Example:

I want to reduce all the prices in my shop by 9% for the January sale. Items cost £1·35, £0·76, etc.

Enter discount [9] [RUN]

Gross price [1] [·] [3] [5] [RUN]

Display shows discounted price £1·23

Gross price [0] [·] [7] [6] [RUN]

Display shows discounted price 69p etc.

(Results shown on display have been rounded to nearest penny.)

—	F	00
÷	G	01
#	3	02
1	1	03
0	0	04
0	0	05
+	E	06
#	3	07
1	1	08
=	—	09
sto	2	10
stop	0	11
X	·	12
rcl	5	13
=	—	14
▼	A	15
goto	2	16
1	1	17
1	1	18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

MARK–UP

÷	G	00
#	3	01
1	1	02
0	0	03
0	0	04
+	E	05
#	3	06
1	1	07
=	–	08
sto	2	09
stop	0	10
X	·	11
rcl	5	12
=	–	13
▼	A	14
goto	2	15
1	1	16
0	0	17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

Marks up a series of prices by a given percentage.

Execution:

percentage mark-up / RUN / price / RUN /
marked up price / another price / RUN /
marked up price / etc.

MARK–UP, GROSS PERCENTAGE INCREASE GIVEN

Marks up prices by a given percentage of their *new* value. Thus £90 marked up by 10% will give £100; the increase of £10 is 10% of the gross price £100.

Execution:

percentage / RUN / old price / RUN / new price / another old price / RUN / new price / etc.

To enter a new percentage:

▲▼ / ▲▼ / goto / 0 / 0 / new percentage / RUN / old price / etc.

÷	G	00
#	3	01
1	1	02
0	0	03
0	0	04
–	F	05
#	3	06
1	1	07
–	F	08
÷	G	09
=	–	10
sto	2	11
stop	0	12
X	·	13
rcl	5	14
=	–	15
▼	A	16
goto	2	17
1	1	18
2	2	19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

DISCOUNT OR TAX, PERCENTAGE OF NET SUM GIVEN

Example:

VAT is at 8%. I price my goods VAT inclusive and wish to work out their net prices.

Execution:

percentage / RUN / gross price / RUN / deduction or tax / RUN / net price / another gross price / RUN / deduction or tax / RUN / net price / etc.

To enter a new percentage:

/ C/CE / C/CE / ▲▼ / ▲▼ / goto / 0 / 0 / new percentage / etc.

÷	G	00
(	6	01
+	E	02
#	3	03
1	1	04
0	0	05
0	0	06
=	–	07
)	6	08
=	–	09
sto	2	10
stop	0	11
–	F	12
(	6	13
X	·	14
rcl	5	15
)	6	16
stop	0	17
=	–	18
▼	A	19
goto	2	20
1	1	21
1	1	22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

PERCENTAGE CHANGE ARISING FROM MARK–UP OR DISCOUNT CHANGE

Example:

VAT is cut from 25% to 12½%. What percentage difference does this make? (By what percentage should prices be cut?)

Execution:

old mark-up / RUN / new mark-up / RUN / percentage change

Enter discounts as negative mark-ups.

Solution to example:

Old mark-up	2 5 RUN
New mark-up	1 2 · 5 RUN

Percentage change = –10%, i.e. 10% decrease.

sto	2	00
÷	G	01
#	3	02
1	1	03
0	0	04
0	0	05
+	E	06
#	3	07
1	1	08
÷	G	09
X	·	10
(	6	11
stop	0	12
–	F	13
rcl	5	14
)	6	15
=	–	16
stop	0	17
▼	A	18
goto	2	19
0	0	20
0	0	21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

MORTGAGE REPAYMENTS

Given:

Amount of mortgage
Length of mortgage
Rate of interest

Finds:

Monthly repayment

Execution:

rate / RUN / term / RUN / amount / RUN / repayment

Example 1:

My mortgage is for a sum of £8500 at 10¾% over 25 years. What must I pay each month?

Rate	1 0 · 7 5 RUN
Term	2 5 RUN
Amount	8 5 0 0 RUN

Monthly repayment = £82·58

Example 2:

My mortgage has 12 years to run. The present balance is £4270. The rate of interest has just been increased to 11%. How much will my new monthly repayment be?

Rate	1 1 RUN
Term	1 2 RUN
Amount	4 2 7 0 RUN

My new monthly payment is £54·81

Note: If you want to work out what your new monthly payment will be following a change of interest rate, and you do not know what your balance is, use one of the programs on page 44 or 45 to calculate your present balance.

÷	G	00
#	3	01
1	1	02
0	0	03
0	0	04
+	E	05
sto	2	06
#	3	07
1	1	08
=	—	09
ln	4	10
X	·	11
stop	0	12
=	—	13
▼	A	14
e^x	4	15
÷	G	16
—	F	17
#	3	18
1	1	19
—	F	20
÷	G	21
rcl	5	22
÷	G	23
stop	0	24
÷	G	25
÷	G	26
#	3	27
1	1	28
2	2	29
=	—	30
stop	0	31
▼	A	32
goto	2	33
0	0	34
0	0	35

BALANCE OUTSTANDING ON A MORTGAGE

Given:

Amount of original mortgage

Monthly repayment

Number of years since mortgage was originally taken out

Rate of interest

Finds:

Balance

Execution:

rate / RUN / number of years / RUN / monthly repayment / RUN / original amount / RUN / balance

Example:

I bought a house seven years ago and took out a mortgage for £5500 at 11½% interest. My monthly repayment has been £70. I now want to sell my house and pay off the mortgage. How much will I have to pay?

Rate	1 1 · 5 RUN
Number of years	7 RUN
Monthly payment	7 0 RUN
Original amount	5 5 0 0 RUN

Balance = £3438

÷	G	00
#	3	01
1	1	02
0	0	03
0	0	04
=	–	05
sto	2	06
+	E	07
#	3	08
1	1	09
=	–	10
ln	4	11
X	·	12
stop	0	13
=	–	14
▼	A	15
e^x	4	16
X	·	17
(	6	18
stop	0	19
X	·	20
#	3	21
1	1	22
2	2	23
÷	G	24
rcl	5	25
=	–	26
sto	2	27
–	F	28
+	E	29
stop	0	30
)	6	31
+	E	32
rcl	5	33
=	–	34
stop	0	35

BALANCE OUTSTANDING ON A MORTGAGE

Given:

Monthly repayments

Present rate of interest

Number of years mortgage has to run

Finds:

Balance outstanding

This program is useful for finding the balance outstanding when the interest rate and/or repayment has changed since the beginning of the mortgage, but the number of years to run is known.

Execution:

interest rate / RUN / number of years to run / RUN / monthly payment / RUN / balance

Example:

My mortgage has 12 years to run. My present monthly payment is £50 and the interest rate is 10½%. What is the outstanding balance?

Rate	1 0 · 5 RUN
Years to run	1 2 RUN
Monthly payment	5 0 RUN

Balance = £3990 to nearest pound.

÷	G	00
#	3	01
1	1	02
0	0	03
0	0	04
+	E	05
sto	2	06
#	3	07
1	1	08
=	–	09
ln	4	10
X	·	11
stop	0	12
–	F	13
=	–	14
▼	A	15
e^x	4	16
–	F	17
#	3	18
1	1	19
–	F	20
X	·	21
stop	0	22
X	·	23
#	3	24
1	1	25
2	2	26
÷	G	27
rcl	5	28
=	–	29
stop	0	30
=	–	31
=	–	32
=	–	33
=	–	34
=	–	35

MORTGAGE TERM

Given:

Amount of mortgage
Monthly payment
Rate of interest

Finds:

Term of mortgage in years

Execution:

rate / RUN / amount of mortgage / RUN / monthly payment / RUN / term

Example 1:

I wish to take out a £7000 mortgage at 11% interest. I can afford to repay £80 per month. What is the shortest term mortgage I can have?

Rate [1] [1] [RUN]

Amount of mortgage [7] [0] [0] [0] [RUN]

Repayment [8] [0] [RUN]

Result is 15·52 years, so in practice I would take out a 15 years mortgage, with a monthly repayment of £81·12 (calculated using the program on page 43).

Example 2:

The balance on my mortgage is £5100 and my monthly repayment is £55. I have just been informed that the interest rate has been increased to 11¼%. I cannot afford a higher repayment and so I shall have to extend the term of the mortgage. When will the mortgage be paid off?

Rate [1] [1] [·] [2] [5]

Amount of mortgage [5] [1] [0] [0]

Repayment [5] [5]

Result is 19·085

So the new term is 19 years with a small balance payable at the end.

÷	G	00
#	3	01
1	1	02
0	0	03
X	·	05
0	0	04
sto	2	06
stop	0	07
÷	G	08
stop	0	09
÷	G	10
#	3	11
1	1	12
2	2	13
–	F	14
#	3	15
1	1	16
–	F	17
÷	G	18
=	–	19
ln	4	20
÷	G	21
(	6	22
rcl	5	23
+	E	24
#	3	25
1	1	26
=	–	27
ln	4	28
)	6	29
=	–	30
stop	0	31
=	–	32
=	–	33
=	–	34
=	–	35

TAX RELIEF ON A MORTGAGE

Given:

Balance of mortgage

Interest rate

Finds:

Annual tax relief (for standard rate taxpayers)

Execution:

balance / RUN / interest rate / RUN /

tax relief

Example:

My mortgage balance is £6000 and the rate of interest is 10¾%. How much tax will I save this year?

Balance $\boxed{6}\ \boxed{0}\ \boxed{0}\ \boxed{0}$

Rate $\boxed{1}\ \boxed{0}\ \boxed{\cdot}\ \boxed{7}\ \boxed{5}$

Tax relief = £225·75

Note: This program assumes tax rate of 35p in the pound. Should this change, the figures in steps 07 and 08 should be altered to correspond.

X	·	00
stop	0	01
X	·	02
#	3	03
·	A	04
0	0	05
0	0	06
3	3	07
5	5	08
=	–	09
stop	0	10
▼	A	11
goto	2	12
0	0	13
0	0	14
		15
		16
		17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

PERIOD RATE TO ANNUAL RATE

(settlement discount and credit cards)

Given:

Interest rate per period

Number of periods per year

Finds:

Equivalent annual rate

Execution:

number of periods per year / RUN / period rate / RUN / annual rate

e.g.

52 / RUN / weekly rate / RUN / annual rate

4 / RUN / quarterly rate / RUN / annual rate

Example:

A car dealer makes a credit agreement with a customer whereby £250 will be paid off in 30 fortnightly instalments of £10. He has used the program on page 54 to calculate that the effective fortnightly rate is 1·195%. Under the Consumer Credit Act, the equivalent annual rate must be specified. What is it?

Number of fortnights per year [2] [6] [RUN]

Fortnightly rate [1] [·] [1] [9] [5] [RUN]

Equivalent annual rate = 36·02%

X	·	00
(	6	01
stop	0	02
÷	G	03
#	3	04
1	1	05
0	0	06
0	0	07
+	E	08
#	3	09
1	1	10
=	–	11
ln	4	12
)	6	13
=	–	14
▼	A	15
e^x	4	16
–	F	17
#	3	18
1	1	19
X	·	20
#	3	21
1	1	22
0	0	23
0	0	24
=	–	25
stop	0	26
▼	A	27
goto	2	28
0	0	29
0	0	30
		31
		32
		33
		34
		35

Settlement discount

Example:

I can claim a discount of 2% if I settle an account due at the end of the month by the 15th of the month. What annual interest rate does this represent?

Solution:

Since months are of unequal lengths, take the period to be 1/2 month or 1/24 year.

Number of periods	[2] [4] [RUN]
Period rate	[2] [RUN]

Annual rate = 60·82% (rounded to nearest ·01%)

Credit Cards

Example:

I must pay 0·5% per week interest on my credit card account. What is the equivalent annual rate?

Number of periods	[5] [2] [RUN]
Period rate	[·] [5] [RUN]

Annual rate = 29·68% (rounded to nearest ·01%)

The same program may be used for calculating the period rate from the annual rate. Use the execution sequence:

number of periods / ÷ / RUN / annual rate / RUN / period rate

Example:

A bank charges 15% interest per annum. What is the equivalent quarterly rate?

Number of periods per year	[4] [÷] [RUN]
Annual rate	[1] [5] [RUN]

Result: Quarterly rate = 3·55% (rounded to nearest ·01%)

DAILY RATE TO ANNUAL RATE

Given:
Daily rate

Finds:
Annual rate

Execution:
daily rate / RUN / annual rate

Note: There is some loss of accuracy for daily rates of above about 0·3%.

X	·	00
#	3	01
3	3	02
·	A	03
6	6	04
5	5	05
=	—	06
▼	A	07
e^x	4	08
—	F	09
#	3	10
1	1	11
X	·	12
#	3	13
1	1	14
0	0	15
0	0	16
=	—	17
stop	0	18
▼	A	19
goto	2	20
0	0	21
0	0	22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

ANNUAL RATE TO DAILY RATE

÷	G	00
#	3	01
1	1	02
0	0	03
0	0	04
+	E	05
#	3	06
1	1	07
=	–	08
ln	4	09
÷	G	10
#	3	11
3	3	12
·	A	13
6	6	14
5	5	15
=	–	16
stop	0	17
▼	A	18
goto	2	19
0	0	20
0	0	21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

Given:
Annual rate

Finds:
Daily rate

Execution:
annual rate / RUN / daily rate

Note: There is some loss of accuracy for annual rates of above about 200%.

MONTHLY RATE TO ANNUAL RATE

Given:

Monthly rate

Finds:

Equivalent annual rate

Comments:

Compounding every month

Execution:

monthly rate / RUN / annual rate

Example:

A dealer has calculated that the monthly interest rate on his H.P. agreements is 1·9%. Under the Consumer Credit Act he must display the annual rate. What is it?

[1] [·] [9] [RUN]

Result 25·32%

÷	G	00
#	3	01
1	1	02
0	0	03
0	0	04
+	E	05
#	3	06
1	1	07
=	—	08
ln	4	09
X	·	10
#	3	11
1	1	12
2	2	13
=	—	14
▼	A	15
e^x	4	16
—	F	17
#	3	18
1	1	19
X	·	20
#	3	21
1	1	22
0	0	23
0	0	24
=	—	25
stop	0	26
▼	A	27
goto	2	28
0	0	29
0	0	30
		31
		32
		33
		34
		35

REGULAR REPAYMENT LOAN

Term of loan

Given:
Amount of loan
Amount of regular repayment
Interest rate

Finds:
Number of repayments

Comments:
Interest compounded every repayment period

Execution:
rate / RUN / amount of loan / RUN / repayment / RUN / number of repayments

Example:
I borrow £1000 at 10% interest. I repay £250 per year. How long will it take to pay off the debt?

Rate	[1] [0] [RUN]
Initial sum	[1] [0] [0] [0] [RUN]
Annual repayment	[2] [5] [0] [RUN]

Answer 5·36 years

In practice I would make 5 payments of £250 and then pay off the balance outstanding; this can be worked out using the program on page 56.

÷	G	00
#	3	01
1	1	02
0	0	03
0	0	04
X	·	05
sto	2	06
stop	0	07
÷	G	08
stop	0	09
—	F	10
#	3	11
1	1	12
—	F	13
÷	G	14
=	—	15
ln	4	16
÷	G	17
(	6	18
rcl	5	19
+	E	20
#	3	21
1	1	22
=	—	23
ln	4	24
)	6	25
=	—	26
stop	0	27
▼	A	28
goto	2	29
0	0	30
0	0	31
		32
		33
		34
		35

REGULAR REPAYMENT LOAN

Interest rate

Given:
Amount of loan
Amount of regular repayments
Number of repayments

Finds:
Interest rate per repayment period

Comments:
Interest compounded each repayment period

Formula:

$$I = \frac{100}{A_o}\left[1 - \frac{1}{\left(1 + \frac{I}{100}\right)^N}\right]$$

Execution:
repayment amount / RUN / amount of loan / RUN / number of repayments / RUN / estimate of rate / RUN / number of repayments / RUN / estimate of rate / RUN / number of repayments / RUN /

keep repeating until two successive values of the estimate of the interest rate are the same; this value is then the required interest rate.

X	·	00
#	3	01
1	1	02
0	0	03
0	0	04
÷	G	05
stop	0	06
÷	G	07
sto	2	08
#	3	09
1	1	10
0	0	11
0	0	12
+	E	13
#	3	14
1	1	15
=	–	16
ln	4	17
X	·	18
stop	0	19
–	F	20
=	–	21
▼	A	22
e^x	4	23
–	F	24
#	3	25
1	1	26
–	F	27
X	·	28
rcl	5	29
÷	G	30
stop	0	31
▼	A	32
goto	2	33
0	0	34
9	9	35

Example:

A television shop sells a £200 television on hire purchase terms of a £50 deposit followed by 18 monthly instalments of £10. Under the Consumer Credit Act, the shop is required to specify what interest rate this represents. What is the effective monthly interest rate?

Solution:

Amount of loan is £200 – £50 = £150

Repayment amount	[1] [0] [RUN]
Amount of loan	[1] [5] [0] [RUN]
Number of repayments	[1] [8] [RUN]
Estimate rate =	[RUN]
	[1] [8] [RUN]
Next estimate =	[RUN]

Repeat until two successive estimates are the same.

After several repetitions, reach the result of 1·9917271%.

Note: to obtain the equivalent annual rate, use the conversion program on page 52.

REGULAR REPAYMENT LOAN

Balance outstanding just after a repayment has been made

Given:

Amount of original loan
Amount of regular repayment
Number of repayments that have been made
Rate of interest per repayment period

Finds:

Amount outstanding

Comments:

Interest compounded each repayment period

Execution:

rate / RUN / number of repayments / RUN / repayment / RUN / original amount / RUN / balance

Example:

I borrowed £500 five years ago at 9% interest. I have repaid £100 each year since then. What will the balance be after this year's payment?

Rate	[9] [RUN]
Number of payments	[5] [RUN]
Payment	[1] [0] [0] [RUN]
Original amount	[5] [0] [0] [RUN]

So I now owe £170·83

÷	G	00
#	3	01
1	1	02
0	0	03
0	0	04
+	E	05
sto	2	06
#	3	07
1	1	08
=	–	09
ln	4	10
X	·	11
stop	0	12
=	–	13
▼	A	14
e^x	4	15
X	·	16
(	6	17
stop	0	18
÷	G	19
rcl	5	20
=	–	21
sto	2	22
–	F	23
+	E	24
stop	0	25
)	6	26
+	E	27
rcl	5	28
=	–	29
stop	0	30
=	–	31
=	–	32
=	–	33
=	–	34
=	–	35

REGULAR REPAYMENT LOAN

Amount of repayment

Given:
Amount of loan
Number of repayment periods
Rate of interest

Finds:
Necessary regular repayment

Comments:
Interest compounded every repayment period

Execution:
rate / RUN / term / RUN / amount of loan / RUN / regular repayment

Example:
I take a loan of £100 at a rate of 1% per month. I want to pay back the money in 36 monthly instalments. How much do I pay per month?

Rate	[1] [RUN]
Term	[3] [6] [RUN]
Amount	[1] [0] [0] [RUN]

Regular repayment = £3·31
(rounded to nearest penny)

÷	G	00
#	3	01
1	1	02
0	0	03
0	0	04
+	E	05
sto	2	06
#	3	07
1	1	08
=	–	09
ln	4	10
X	·	11
stop	0	12
=	–	13
▼	A	14
e^x	4	15
÷	G	16
–	F	17
#	3	18
1	1	19
–	F	20
÷	G	21
rcl	5	22
÷	G	23
stop	0	24
÷	G	25
=	–	26
stop	0	27
▼	A	28
goto	2	29
0	0	30
0	0	31
		32
		33
		34
		35

SINGLE REPAYMENT LOAN

Final amount

Given:

Rate of interest per accounting period
Number of accounting periods
Initial sum

To find:

Final sum

Comments:

Interest compounded each accounting period

Execution:

rate of interest / RUN / number of periods / RUN / initial sum / RUN / final sum

Formula:

$$F = I\left(1 + \frac{\alpha}{100}\right)^n$$

÷	G	00
#	3	01
1	1	02
0	0	03
0	0	04
+	E	05
#	3	06
1	1	07
=	–	08
ln	4	09
X	·	10
stop	0	11
=	–	12
▼	A	13
e^x	4	14
X	·	15
stop	0	16
=	–	17
stop	0	18
▼	A	19
goto	2	20
0	0	21
0	0	22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

SINGLE REPAYMENT LOAN

Final amount

Given:
Annual rate of interest
Term of loan
Initial sum

To find:
Final sum

Comments:
Interest compounded every six months

Execution:
rate of interest / RUN / term in years / RUN / initial sum / RUN / final sum

Example:
I invest £570 at 8% interest. How much is in my account after 5 years?

Rate of interest	8 RUN
Term in years	5 RUN
Initial sum	5 7 0 RUN

Answer £843·65

Formula:

$$F = I\left(1 + \frac{a}{100}\right)^{2n}$$

÷	G	00
#	3	01
2	2	02
0	0	03
0	0	04
+	E	05
#	3	06
1	1	07
=	–	08
ln	4	09
X	·	10
stop	0	11
+	E	12
=	–	13
▼	A	14
e^x	4	15
X	·	16
stop	0	17
=	–	18
stop	0	19
▼	A	20
goto	2	21
0	0	22
0	0	23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

SINGLE REPAYMENT LOAN

Number of years to achieve given result

Given:
Initial sum
Final sum
Rate of interest per accounting period

Finds:
Number of accounting periods

Comments:
Interest compounded each accounting period

Execution:
rate / RUN / initial sum / RUN / final sum / RUN / term

Example:
How long will it take £700 to become £2000 if interest of 12½% is paid annually?

Rate	[1] [2] [·] [5] [RUN]
Initial sum	[7] [0] [0] [RUN]
Final sum	[2] [0] [0] [0] [RUN]

Answer 8·916 years; so the first time the balance will exceed £2000 will be after the ninth interest payment.

÷	G	00
#	3	01
1	1	02
0	0	03
0	0	04
+	E	05
#	3	06
1	1	07
=	–	08
ln	4	09
sto	2	10
stop	0	11
÷	G	12
stop	0	13
÷	G	14
=	–	15
ln	4	16
÷	G	17
rcl	5	18
=	–	19
stop	0	20
▼	A	21
goto	2	22
0	0	23
0	0	24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

SINGLE REPAYMENT LOAN

Number of years to achieve given result

Given:
Initial sum
Final sum
Annual rate of interest

Finds:
Term

Comments:
Interest compounded every six months

Execution:
rate / RUN / initial sum / RUN / final sum / RUN / term

÷	G	00
#	3	01
2	2	02
0	0	03
0	0	04
+	E	05
#	3	06
1	1	07
=	–	08
ln	4	09
sto	2	10
stop	0	11
÷	G	12
stop	0	13
÷	G	14
=	–	15
ln	4	16
÷	G	17
rcl	5	18
÷	G	19
#	3	20
2	2	21
=	–	22
stop	0	23
▼	A	24
goto	2	25
0	0	26
0	0	27
		28
		29
		30
		31
		32
		33
		34
		35

SINGLE REPAYMENT LOAN

Interest rate needed for given result

Given:

Number of accounting periods

Initial and final sum

Finds:

Effective rate of interest per accounting period

Comments:

Interest compounded every accounting period

Execution:

initial sum / RUN / final sum / RUN / term / RUN / rate of interest

÷	G	00
stop	0	01
÷	G	02
=	—	03
ln	4	04
÷	G	05
stop	0	06
=	—	07
▼	A	08
e^x	4	09
—	F	10
#	3	11
1	1	12
X	·	13
#	3	14
1	1	15
0	0	16
0	0	17
=	—	18
stop	0	19
▼	A	20
goto	2	21
0	0	22
0	0	23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

SINGLE REPAYMENT LOAN

Interest rate for given result

Given:
Term in years
Initial and final sum

Finds:
Effective annual interest rate

Comments:
Interest compounded every six months

Execution:
initial sum / RUN / final sum / RUN / term / RUN / rate of interest

Example:
A bond costs £100 and is repayable in 4 years at £150. What rate of interest does this represent?

Initial sum	1 0 0 RUN
Final sum	1 5 0 RUN
Term	4 RUN

Equivalent interest rate = 10·38%

÷	G	00
stop	0	01
÷	G	02
=	–	03
ln	4	04
÷	G	05
(	6	06
stop	0	07
+	E	08
)	6	09
=	–	10
▼	A	11
e^x	4	12
–	F	13
#	3	14
1	1	15
X	·	16
#	3	17
2	2	18
0	0	19
0	0	20
=	–	21
stop	0	22
▼	A	23
goto	2	24
0	0	25
0	0	26
		27
		28
		29
		30
		31
		32
		33
		34
		35

PRESENT VALUE OF A SINGLE FUTURE PAYMENT

Given:

Rate of interest per accounting period

Number of periods ahead that payment is to be made

Finds:

Present value of future payment

Comments:

Interest compounded every accounting period

Execution:

rate / RUN / term / RUN / amount / RUN / present value

Formula:

$$I = \frac{F}{\left(1 + \frac{a}{100}\right)^n}$$

÷	G	00
#	3	01
1	1	02
0	0	03
0	0	04
+	E	05
#	3	06
1	1	07
÷	G	08
=	–	09
ln	4	10
X	·	11
stop	0	12
=	–	13
▼	A	14
e^x	4	15
X	·	16
stop	0	17
=	–	18
stop	0	19
▼	A	20
goto	2	21
0	0	22
0	0	23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

PRESENT VALUE OF A SINGLE FUTURE PAYMENT

Given:

Annual rate of interest

Number of years ahead that payment is to be made

Amount of payment

Finds:

Present value

Comments:

Interest compounded every six months

Execution:

rate / RUN / term / RUN / amount / RUN / present value

Example:

What is the present value of a payment of £5000 made in 4 years time at an annual rate of 14%?

Rate [1] [4] [RUN]

Term [4] [RUN]

Amount [5] [0] [0] [0] [RUN]

Answer: present value = £2909·67

Formula:

$$I = \frac{F}{\left(1 + \frac{a}{200}\right)^{2n}}$$

÷	G	00
#	3	01
2	2	02
0	0	03
0	0	04
+	E	05
#	3	06
1	1	07
÷	G	08
=	–	09
ln	4	10
X	·	11
sto	2	12
stop	0	13
+	E	14
=	–	15
▼	A	16
e^x	4	17
X	·	18
stop	0	19
=	–	20
stop	0	21
▼	A	22
goto	2	23
0	0	24
0	0	25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

PRESENT VALUE OF A SERIES OF POSSIBLE UNEQUAL FUTURE PAYMENTS

Given:

Payments

Interest rate per payment period

Finds:

Present value

Execution:

Suppose payments are made of p_1 at the end of the first year, p_2 at the end of the second year, and so on up to a final payment of p_n at the end of the nth year.

Use the following execution sequence:
interest rate / RUN / p_n / RUN / · · · / RUN / p_1 / RUN / present value of all future payments

Before a new calculation:

C/CE / ▲▼ / ▲▼ / goto / 0 / 0 /

Notice that the payments are entered *in reverse order*, with the last payment first.

Example:

An investor wishes to make future payments to a businessman as follows:

1 Jan.	1978	£10,000
	1979	£12,000
	1980	£15,000
	1981	£20,000
	1982	£20,000

÷	G	00
#	3	01
1	1	02
0	0	03
0	0	04
+	E	05
#	3	06
1	1	07
÷	G	08
=	–	09
sto	2	10
stop	0	11
X	·	12
rcl	5	13
+	E	14
▼	A	15
goto	2	16
1	1	17
1	1	18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

Reckoning the annual interest rate to be 14%, what is the value of these payments on 1 Jan. 1977?

Rate				1	4	RUN
1982	2	0	0	0	0	RUN
1981	2	0	0	0	0	RUN
1980	1	5	0	0	0	RUN
1979	1	2	0	0	0	RUN
1978	1	0	0	0	0	RUN

Payments in reverse order

Present value = £50,359

PRESENT VALUE OF A SERIES OF EQUAL FUTURE PAYMENTS

Given:

Rate of interest per payment period
Number of payments
Amount of each payment

Finds:

Present value

Comments:

Assumes payments start at the end of the first payment period
Interest compounded each payment period

Execution:

rate / RUN / number of payments / RUN / amount of each payment / RUN / present value

Example:

Find the present value of £200,000 paid in 20 equal annual instalments. The rate of interest is 13% and the first payment is made immediately.

Solution:

There are 19 equal future payments of £10,000 and one present payment. Find the present value of the future payments first and then add the present payment.

Rate	1 3 RUN
Number of payments	1 9 RUN
Amount	1 0 0 0 0 RUN
Add present payment	+ 1 0 0 0 0 =

So present value of all payments is £79,379

÷	G	00
#	3	01
1	1	02
0	0	03
0	0	04
+	E	05
sto	2	06
#	3	07
1	1	08
=	–	09
ln	4	10
X	·	11
stop	0	12
–	F	13
=	–	14
▼	A	15
e^x	4	16
–	F	17
#	3	18
1	1	19
–	F	20
÷	G	21
rcl	5	22
X	·	23
stop	0	24
=	–	25
stop	0	26
▼	A	27
goto	2	28
0	0	29
0	0	30
		31
		32
		33
		34
		35

PRESENT VALUE OF A SERIES OF EQUAL PAYMENTS FOLLOWED BY A SINGLE PAYMENT

(e.g. Dated government stocks)

Given:

Regular payment (paid at the end of each repayment period including the last)

Final payment (excluding final regular payment)

Number of repayment periods

Discounting interest rate per repayment period

Finds:

Present value of future payments

Comments:

Notional interest compounded each repayment period.

Execution:

interest rate / RUN / final payment / RUN / present value

Example:

What is the present value of a government stock which yields £35 every half year and will be repaid at £1000 in 8½ years time? Take interest rate for discounting to be 6½% per half year.

Rate	6 · 5 RUN
Number of repayments	1 9 RUN
Regular payment	3 5 RUN
Final payment	1 0 0 0 RUN

Present value = £677·96

÷	G	00
#	3	01
1	1	02
0	0	03
0	0	04
+	E	05
sto	2	06
#	3	07
1	1	08
=	–	09
ln	4	10
X	·	11
stop	0	12
–	F	13
=	–	14
▼	A	15
e^x	4	16
▼	A	17
MEx	5	18
÷	G	19
X	·	20
stop	0	21
=	–	22
▼	A	23
MEx	5	24
X	·	25
(	6	26
stop	0	27
–	F	28
rcl	5	29
)	6	30
+	E	31
rcl	5	32
=	–	33
stop	0	34
=	–	35

		00
		01
		02
		03
		04
		05
		06
		07
		08
		09
		10
		11
		12
		13
		14
		15
		16
		17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

		00
		01
		02
		03
		04
		05
		06
		07
		08
		09
		10
		11
		12
		13
		14
		15
		16
		17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

MEAN AND STANDARD DEVIATION

Observations $x_1, \cdots, x_n$

Mean $\bar{x} = \frac{1}{n}\Sigma x_i$

(i) Standard deviation about mean

$$\sigma = \sqrt{\frac{1}{n}\Sigma(x_i - \bar{x})^2}$$

(ii) Standard deviation about a

$$\sigma_a = \sqrt{\frac{1}{n}\Sigma(x_i - a)^2}$$

Execution:

(i) RUN / x_1 / RUN / x_2 / $\cdots$ / x_n / RUN / ▲▼ / ▲▼ / goto / 1 / 9 / RUN * / n / RUN / $\bar{x}$ / RUN / σ

(ii) as (i) to *, then $\cdots$ / n / RUN / $\bar{x}$ / a / RUN / σ_a

#	3	00
0	0	01
=	—	02
sto	2	03
(	6	04
stop	0	05
+	E	06
▼	A	07
MEx	5	08
=	—	09
▼	A	10
MEx	5	11
X	·	12
)	6	13
+	E	14
▼	A	15
goto	2	16
0	0	17
4	4	18
rcl	5	19
÷	G	20
stop	0	21
sto	2	22
=	—	23
stop	0	24
X	·	25
X	·	26
rcl	5	27
—	F	28
)	6	29
÷	G	30
rcl	5	31
=	—	32
$\sqrt{x}$	1	33
stop	0	34
=	—	35

MEAN, SUM OF SQUARES ABOUT MEAN, AND ESTIMATE OF VARIANCE

Mean $\bar{x} = \frac{1}{n}\Sigma x_i$

Sum of squares about mean $S_{xx} = \Sigma(x_i - \bar{x})^2$

Estimate of variance $s^2 = \frac{S_{xx}}{n-1}$

Pre-execution:

Before each set of data is entered, clear memory with / C/CE / ▲▼ / sto /

Execution:

RUN / x_1 / RUN / x_2 / · · · / x_n / RUN / Σx^2 / ▲▼ / ▲▼ / goto / 1 / 5 / RUN / Σx / n / RUN / $\bar{x}$ / RUN / S_{xx} / RUN / s^2

(	6	00
stop	0	01
+	E	02
▼	A	03
MEx	5	04
=	–	05
▼	A	06
MEx	5	07
X	·	08
)	6	09
+	E	10
▼	A	11
goto	2	12
0	0	13
0	0	14
rcl	5	15
÷	G	16
stop	0	17
sto	2	18
X	·	19
stop	0	20
X	·	21
rcl	5	22
–	F	23
)	6	24
÷	G	25
stop	0	26
(	6	27
rcl	5	28
–	F	29
#	3	30
1	1	31
=	–	32
)	6	33
=	–	34
stop	0	35

LINEAR REGRESSION AND CORRELATION COEFFICIENT

sto	2	00
(	6	01
stop	0	02
–	F	03
rcl	5	04
×	·	05
stop	0	06
)	6	07
+	E	08
▼	A	09
goto	2	10
0	0	11
1	1	12
÷	G	13
(	6	14
stop	0	15
÷	G	16
stop	0	17
sto	2	18
=	–	19
$\sqrt{x}$	1	20
×	·	21
▼	A	22
MEx	5	23
=	–	24
)	6	25
÷	G	26
stop	0	27
rcl	5	28
×	·	29
stop	0	30
–	F	31
stop	0	32
–	F	33
=	–	34
stop	0	35

Observations $(x_1, y_1), \cdots, (x_n, y_n)$

Sum of cross products $S_{xy} = \Sigma(x_i - \bar{x})(y_i - \bar{y})$

Correlation coefficient

$$r = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Sigma(x_i - \bar{x})^2 \Sigma(y_i - \bar{y})^2}}$$

Regression line (y on x) $y = a + bx$

Method:

First use program on page 73 applied to the x's and y's separately to calculate $\bar{x}$, S_{xx}, $\bar{y}$ and S_{yy}. Then use this program as follows.

Execution:

$\bar{x}$ / RUN / x_1 / RUN / y_1 / RUN / x_2 / RUN / y_2 / · · · / x_n / RUN / y_n / ▲▼ /) / = / ▲▼ / ▲▼ / goto / 1 / 3 / RUN / / S_{xx} / RUN / S_{yy} / RUN / / RUN / / $\bar{x}$ / RUN / $\bar{y}$ / RUN /

SLOPE OF REGRESSION LINE

(	6	00
stop	0	01
+	E	02
▼	A	03
MEx	5	04
=	–	05
▼	A	06
MEx	5	07
X	·	08
stop	0	09
)	6	10
+	E	11
▼	A	12
goto	2	13
0	0	14
0	0	15
(	6	16
rcl	5	17
–	F	18
÷	G	19
stop	0	20
X	·	21
▼	A	22
MEx	5	23
)	6	24
÷	G	25
X	·	26
(	6	27
rcl	5	28
X	·	29
stop	0	30
+	E	31
stop	0	32
)	6	33
=	–	34
stop	0	35

Regression line is $y = a + bx$

Observations $(x_1, y_1), (x_2, y_2), \cdots (x_n, y_n)$

Execution:

RUN / y_1 / RUN / x_1 / RUN / y_2 / RUN / x_2 / RUN / · · · / x_n / RUN / Σxy (note) / C/CE / ▲▼ / ▲▼ / MEx / Σy (note) / ▲▼ / (/ x_1 / RUN / RUN / x_2 / RUN / RUN / · · · / x_n / RUN / ▲▼ /) / + / ▲▼ / ▲▼ / goto / 1 / 6 / RUN / n / RUN / Σy / RUN / Σxy / RUN / b

Note: The values of Σxy and Σy must be written down and re-entered later in the execution sequence.

TESTING THE HYPOTHESIS OF ZERO CORRELATION

Assuming normality, on the hypothesis that $\rho = 0$, the statistic

$$t = r\,\frac{\sqrt{N-2}}{\sqrt{1-r^2}}$$

has the t distribution with (N – 2) degrees of freedom. Large values of t indicate that the true correlation coefficient is non-zero.

Execution:

r / RUN / N / RUN /

÷	G	00
(	6	01
X	·	02
–	F	03
#	3	04
1	1	05
–	F	06
=	–	07
$\sqrt{x}$	1	08
)	6	09
X	·	10
(	6	11
stop	0	12
–	F	13
#	3	14
2	2	15
=	–	16
$\sqrt{x}$	1	17
)	6	18
=	–	19
stop	0	20
▼	A	21
goto	2	22
0	0	23
0	0	24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

REGRESSION LINE SLOPE

To test whether it is significantly different from zero or any other given value b_0

Slope of regression line = b

Correlation coefficient = r

Sample size = N

Calculate the statistic

$$t = \frac{(b - b_0)\sqrt{N - 2}}{\sqrt{1 - r^2}}$$

On the null hypothesis that the true value of b is b_0, this has the t-distribution with (N − 2) degrees of freedom (approximately standard normal if N is reasonably large).

Execution:

b_0 / RUN / b / RUN / r / RUN / N / RUN / t

If b_0 is zero the following can be used:

b / RUN / RUN / r / RUN / N / RUN / t

–	F	00
stop	0	01
–	F	02
÷	G	03
(	6	04
stop	0	05
X	·	06
–	F	07
#	3	08
1	1	09
–	F	10
=	–	11
$\sqrt{x}$	1	12
)	6	13
X	·	14
(	6	15
stop	0	16
–	F	17
#	3	18
2	2	19
=	–	20
$\sqrt{x}$	1	21
)	6	22
=	–	23
stop	0	24
▼	A	25
goto	2	26
0	0	27
0	0	28
		29
		30
		31
		32
		33
		34
		35

STUDENT'S t–TEST

$$t = \frac{\bar{x}\sqrt{n}}{s}$$

To test whether the mean of a set of observations $x_1, \cdots, x_n$ differs significantly from zero. Large values of t reject the hypothesis that the mean is zero.

Pre-execution:

Clear memory with C/CE / ▲▼ / sto /

Execution:

RUN / x_1 / RUN / x_2 / · · · / x_n / RUN / ▲▼ / ▲▼ / goto / 1 / 5 / RUN / n / RUN / n / RUN / t

To re-use:

C/CE / ▲▼ / sto / ▲▼ / ▲▼ / goto / 0 / 0 /

(	6	00
stop	0	01
+	E	02
▼	A	03
MEx	5	04
=	–	05
▼	A	06
MEx	5	07
X	·	08
)	6	09
+	E	10
▼	A	11
goto	2	12
0	0	13
0	0	14
rcl	5	15
X	·	16
÷	G	17
stop	0	18
–	F	19
)	6	20
÷	G	21
(	6	22
stop	0	23
÷	G	24
–	F	25
#	3	26
1	1	27
–	F	28
)	6	29
=	–	30
$\sqrt{x}$	1	31
÷	G	32
X	·	33
rcl	5	34
=	–	35

STUDENT'S t–TEST

To test whether the mean is significantly different from some value a:

$$t = \frac{(\bar{x} - a)\sqrt{n}}{s}$$

Pre-execution (before each set of data):

/ ▲▼ / ▲▼ / goto / 0 / 0 / C/CE / C/CE / ▲▼ / sto /

Execution:

RUN / x_1 / RUN / x_2 / · · · / x_n / RUN / ▲▼ / ▲▼ / goto / 1 / 5 / RUN / n / RUN / n / RUN / n – 1 / RUN / a / RUN / = /

(	6	00
stop	0	01
+	E	02
▼	A	03
MEx	5	04
=	–	05
▼	A	06
MEx	5	07
×	·	08
)	6	09
+	E	10
▼	A	11
goto	2	12
0	0	13
0	0	14
rcl	5	15
÷	G	16
stop	0	17
×	·	18
▼	A	19
MEx	5	20
–	F	21
)	6	22
÷	G	23
×	·	24
stop	0	25
×	·	26
stop	0	27
=	–	28
$\sqrt{x}$	1	29
×	·	30
(	6	31
rcl	5	32
–	F	33
stop	0	34
=	–	35

CHI–SQUARED

(	6	00
stop	0	01
–	F	02
stop	0	03
sto	2	04
×	·	05
÷	G	06
rcl	5	07
)	6	08
+	E	09
▼	A	10
goto	2	11
0	0	12
0	0	13
		14
		15
		16
		17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

Observed values $O_1, \cdots, O_n$

Expected values $E_1, \cdots, E_n$

$$\chi^2 = \Sigma \frac{(O_i - E_i)^2}{E_i}$$

Execution:

RUN / O_1 / RUN / E_1 / RUN / O_2 / RUN / E_2 / $\cdots$ / O_n / RUN / E_n / RUN / χ^2

For new data:

Clear with C/CE / C/CE / ▲▼ / ▲▼ / goto / 0 / 0 /

CHI–SQUARED WITH YATES CORRECTION

(e.g. for small contingency tables)

$$\chi^2 = \Sigma \frac{(|O_i - E_i| - \frac{1}{2})^2}{E_i}$$

Execution:

RUN / O_1 / RUN / E_1 / RUN / O_2 / RUN / E_2 / · · · / O_n / RUN / E_n / RUN / χ^2

(	6	00
stop	0	01
–	F	02
stop	0	03
sto	2	04
X	·	05
=	–	06
$\sqrt{x}$	1	07
–	F	08
#	3	09
·	A	10
5	5	11
X	·	12
)	6	13
+	E	14
▼	A	15
goto	2	16
0	0	17
0	0	18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

TWO SAMPLE CHI–SQUARED

$$\chi^2 = \Sigma \frac{(O_i - O_i')^2}{O_i + O_i'}$$

Pre-execution:

Clear memory with C/CE / ▲▼ / sto /

Execution:

O_1 / RUN / O_1' / RUN / O_1 / RUN / O_2 / RUN / O_2' / RUN / O_2 / · · · / O_n / RUN / O_n' / RUN / O_n / RUN /

+	E	00
stop	0	01
÷	G	02
X	·	03
(	6	04
+	E	05
÷	G	06
–	F	07
stop	0	08
+	E	09
X	·	10
)	6	11
+	E	12
rcl	5	13
=	–	14
sto	2	15
stop	0	16
▼	A	17
goto	2	18
0	0	19
0	0	20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

TWO SAMPLE CHI–SQUARED WITH YATES CORRECTION

$$\chi^2 = \Sigma \frac{(|O_i - O_i'| - 1)^2}{O_i + O_i'}$$

Execution:

O_1 / RUN / O_1' / RUN / O_1 / RUN / O_2 / RUN / O_2' / RUN / O_2 / · · · / O_n / RUN / O_n' / RUN / O_n / RUN / χ^2

Caution:

If for any j, $O_j = O_j' = 0$, do not enter either of them but go straight on to O_{j+1}. In any case it is not very sound statistically to use the χ^2 if any of the $(O_i + O_i')$ are less than about 10.

+	E	00
stop	0	01
÷	G	02
X	·	03
(	6	04
+	E	05
÷	G	06
–	F	07
stop	0	08
+	E	09
X	·	10
=	–	11
$\sqrt{x}$	1	12
–	F	13
#	3	14
1	1	15
X	·	16
)	6	17
+	E	18
rcl	5	19
=	–	20
sto	2	21
stop	0	22
▼	A	23
goto	2	24
0	0	25
0	0	26
		27
		28
		29
		30
		31
		32
		33
		34
		35

CONTINGENCY TABLE: χ^2 –TEST FOR INDEPENDENCE

Given a contingency table with h rows and k columns, and observation O_{ij} at the intersection of the ith row and the jth column, it is often of interest to test whether the 'row effect' and 'column effect' are independent. To do this, proceed as follows:

1. Work out the row totals R_i, the column totals C_j and the grand total N.
2. Use the program opposite to calculate the expected values E_{ij} for each cell in the table.
3. Use one of the one-sample χ^2 programs above to work out the χ^2 statistic defined by

$$\Sigma \frac{(O-E)^2}{E} \qquad \text{or} \qquad \Sigma \frac{(|O-E|-\frac{1}{2})^2}{E}$$

Make sure that the observed and expected values are entered for every cell of the table. Use the Yates corrected version if the table is small. The number of degrees of freedom is $(h-1)(k-1)$. If this is fairly large the resulting statistic may be transformed to have a standard normal distribution on the hypothesis of independence by using the transformation program on page 93.

CALCULATING THE EXPECTED VALUES IN A CONTINGENCY TABLE

$$E_{ij} = \frac{R_i C_j}{n}$$

Execution:

N / RUN / R_1 / RUN / C_1 / RUN / E_{11} / RUN / RUN / C_2 / RUN / E_{12} / · · · / [illegible] / RUN / R_2 / RUN / C_1 / RUN / E_{21} / · · · etc.

The current row total is displayed between the two successive / RUN / steps after each result is displayed. It should be altered at this point when moving on from one row to the next.

sto	2	00
stop	0	01
+	E	02
(	6	03
X	·	04
stop	0	05
÷	G	06
rcl	5	07
=	–	08
stop	0	09
#	3	10
0	0	11
=	–	12
)	6	13
=	–	14
▼	A	15
goto	2	16
0	0	17
1	1	18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

Z STATISTIC

For testing whether a proportion is significantly different from θ. The statistic Z has mean 0 and variance 1 and is approximately normally distributed.

$$Z = \frac{\frac{x}{n} - \theta}{\sqrt{\frac{\theta(1-\theta)}{n}}}$$

Execution:

θ / RUN / n / RUN / x / RUN /

sto	2	00
—	F	01
(	6	02
X	·	03
)	6	04
=	—	05
$\sqrt{x}$	1	06
÷	G	07
X	·	08
(	6	09
rcl	5	10
X	·	11
stop	0	12
sto	2	13
—	F	14
stop	0	15
—	F	16
)	6	17
÷	G	18
(	6	19
rcl	5	20
$\sqrt{x}$	1	21
)	6	22
=	—	23
stop	0	24
▼	A	25
goto	2	26
0	0	27
0	0	28
		29
		30
		31
		32
		33
		34
		35

NON-PARAMETRIC STATISTICS

Spearman's rank correlation coefficient

Pairs of ranks
$(r_1, s_1), (r_2, s_2), \cdots, (r_n, s_n)$

Execution:

n / RUN / r_1 / RUN / s_1 / RUN / $\cdots$ / r_n / RUN / s_n / RUN / ρ

sto	2	00
X	·	01
X	·	02
rcl	5	03
–	F	04
rcl	5	05
–	F	06
÷	G	07
#	3	08
6	6	09
=	–	10
sto	2	11
#	3	12
1	1	13
+	E	14
(	6	15
stop	0	16
–	F	17
stop	0	18
X	·	19
÷	G	20
rcl	5	21
)	6	22
▼	A	23
goto	2	24
1	1	25
4	4	26
		27
		28
		29
		30
		31
		32
		33
		34
		35

QUALITY CONTROL

Action and warning limits for proportion of batch having given attribute.

$$a\pm = p \pm \alpha\sqrt{\frac{p(1-p)}{n}}$$

Typical values of α:

For action limits $\alpha = 3{\cdot}12$

For warning limits $\alpha = 1{\cdot}96$

Execution:

p / RUN / n / RUN / α / RUN / a− / RUN / a+

sto	2	00
−	F	01
(	6	02
×	·	03
)	6	04
÷	G	05
stop	0	06
=	−	07
$\sqrt{x}$	1	08
×	·	09
stop	0	10
=	−	11
▼	A	12
MEx	5	13
−	F	14
rcl	5	15
+	E	16
stop	0	17
rcl	5	18
+	E	19
rcl	5	20
=	−	21
stop	0	22
▼	A	23
goto	2	24
0	0	25
0	0	26
		27
		28
		29
		30
		31
		32
		33
		34
		35

NORMAL DENSITY FUNCTION

$$\phi = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

Execution:

x / RUN / μ / RUN / σ / RUN / ϕ

–	F	00
stop	0	01
÷	G	02
stop	0	03
sto	2	04
X	·	05
–	F	06
=	–	07
▼	A	08
e^x	4	09
÷	G	10
#	3	11
6	6	12
·	A	13
2	2	14
8	8	15
3	3	16
1	1	17
9	9	18
=	–	19
$\sqrt{x}$	1	20
÷	G	21
rcl	5	22
=	–	23
stop	0	24
▼	A	25
goto	2	26
0	0	27
0	0	28
		29
		30
		31
		32
		33
		34
		35

PERCENTAGE POINTS OF THE NORMAL DISTRIBUTION

Given any α with $0 < \alpha < 0{\cdot}5$, finds x to within about 2 sig. fig. so that the probability that a standard normal random variable exceeds x is α.

Execution:

α / RUN /

For greater accuracy (·1% error) divide result by 1·006.

For still greater accuracy use execution sequence

α / X / 1·0007 / RUN / ÷ / 1·006 / = /

X	·	00
÷	G	01
=	—	02
ln	4	03
$\sqrt{x}$	1	04
sto	2	05
+	E	06
+	E	07
+	E	08
#	3	09
1	1	10
2	2	11
·	A	12
5	5	13
÷	G	14
(	6	15
rcl	5	16
+	E	17
#	3	18
7	7	19
X	·	20
rcl	5	21
+	E	22
#	3	23
5	5	24
=	—	25
)	6	26
—	F	27
+	E	28
rcl	5	29
=	—	30
stop	0	31
▼	A	32
goto	2	33
0	0	34
0	0	35

POISSON DISTRIBUTION

Suppose a random variable has the Poisson distribution with parameter λ. What is the probability that the random variable takes the value j?

Formula:

$$\text{prob}\ (j) = \frac{e^{-\lambda}\lambda^{j}}{j!}$$

Execution:

λ / RUN / j / RUN / [illegible]

Note: Long execution times are possible for large values of j.

–	F	00
(	6	01
ln	4	02
X	·	03
stop	0	04
sto	2	05
)	6	06
–	F	07
–	F	08
(	6	09
rcl	5	10
–	F	11
#	3	12
1	1	13
+	E	14
▼	A	15
gin	1	16
2	2	17
9	9	18
sto	2	19
#	3	20
1	1	21
=	–	22
ln	4	23
)	6	24
▼	A	25
goto	2	26
0	0	27
8	8	28
=	–	29
rcl	5	30
)	6	31
=	–	32
▼	A	33
e^{x}	4	34
stop	0	35

FISHER'S Z TRANSFORMATION FOR CORRELATION COEFFICIENTS.

$$z = \tfrac{1}{2}\log\left(\frac{1+\rho}{1-\rho}\right)$$

The distribution of z is approximately normal.

Execution:

ρ / RUN / [illegible] / n / RUN / σ

where n is the sample size and σ is the standard deviation of z.

$$\sigma = \frac{1}{\sqrt{n-3}}$$

–	F	00
#	3	01
1	1	02
÷	G	03
+	E	04
+	E	05
#	3	06
1	1	07
–	F	08
=	–	09
$\sqrt{x}$	1	10
ln	4	11
stop	0	12
–	F	13
#	3	14
3	3	15
÷	G	16
=	–	17
$\sqrt{x}$	1	18
stop	0	19
▼	A	20
goto	2	21
0	0	22
0	0	23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

TRANSFORMING χ^2 TO NORMAL

X	·	00
+	E	01
=	—	02
$\sqrt{x}$	1	03
—	F	04
(	6	05
stop	0	06
+	E	07
—	F	08
#	3	09
1	1	10
=	—	11
$\sqrt{x}$	1	12
)	6	13
=	—	14
stop	0	15
▼	A	16
goto	2	17
0	0	18
0	0	19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

Suppose x has the χ^2 distribution with n degrees of freedom, where n is fairly large (say $n \geqslant 20$).

Then $y = \sqrt{2x^2} - \sqrt{2n-1}$ has approximately a standard normal distribution with mean 0 and variance 1.

Execution:

x / RUN / n / RUN / y

TRANSFORMING BINOMIAL TO NORMAL

Suppose x is binomially distributed with parameters n and p. Then

$$z = \frac{\frac{x}{n} - p}{\sqrt{\frac{p(1-p)}{n}}}$$

has very nearly a standard normal distribution provided np and n(1 – p) are both greater than 5.

Execution:

p / RUN / n / RUN / x / RUN / z

sto	2	00
–	F	01
(	6	02
X	·	03
)	6	04
=	–	05
$\sqrt{x}$	1	06
÷	G	07
X	·	08
(	6	09
rcl	5	10
X	·	11
stop	0	12
sto	2	13
–	F	14
stop	0	15
–	F	16
)	6	17
÷	G	18
(	6	19
rcl	5	20
$\sqrt{x}$	1	21
)	6	22
=	–	23
stop	0	24
▼	A	25
goto	2	26
0	0	27
0	0	28
		29
		30
		31
		32
		33
		34
		35

		00
		01
		02
		03
		04
		05
		06
		07
		08
		09
		10
		11
		12
		13
		14
		15
		16
		17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

		00
		01
		02
		03
		04
		05
		06
		07
		08
		09
		10
		11
		12
		13
		14
		15
		16
		17
		18
		19
		20
		21
		22
		23
		24
		25
		26
		27
		28
		29
		30
		31
		32
		33
		34
		35

London Rd
St Ives
Huntingdon
Cambs PE17 4HJ
part no. 48584 351